RESEARCH REVIEW 2019

Causal Models for Software Cost Prediction & Control (SCOPE)

Dr. Michael Konrad, Robert Stoddard, Dr. William Nichols, and Dr. David Zubrow

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

Why Causal Learning?

Estimating and controlling program costs would benefit from **causal** knowledge of program dynamics.

Regression does not **distinguish** between correlation and causation.

Causal knowledge is **actionable** knowledge.

Causal discovery is now **practical** and supported with **innovative** tools and algorithms.

Establishing causation with observational data remains a vital need and a key technical challenge but is becoming more feasible and practical.

© 2019 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

Contrary and Surprising Results

Many different types of **complexity** are thought to affect program **success**.

- But the only consistent driver of success or failure we've found is **cognitive fog**.

The number of Information Assurance Vulnerability Alerts (IAVAs) addressed per month was thought to drive IAVA-release **effort**.

- But the most persistent drivers of such effort are **funding factors**.
- When controlling for super domain (SD), the relationship between IAVAs and effort **disappears**.

On the basis of earlier work, it was found that architecture pattern violations did not introduce **security vulnerabilities**.

- But a causal analysis discovered the contrary: **architecture pattern violations** do drive **security vulnerabilities**.

© 2019 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

What Types of Complexity Drive/Impede Project Success?

In 2012, Sheard found that 3 of 40 measures of complexity correlated highly with 7 measures of success:

1) difficult requirements
2) stakeholder relationships
3) cognitive fog

But **causal learning** found

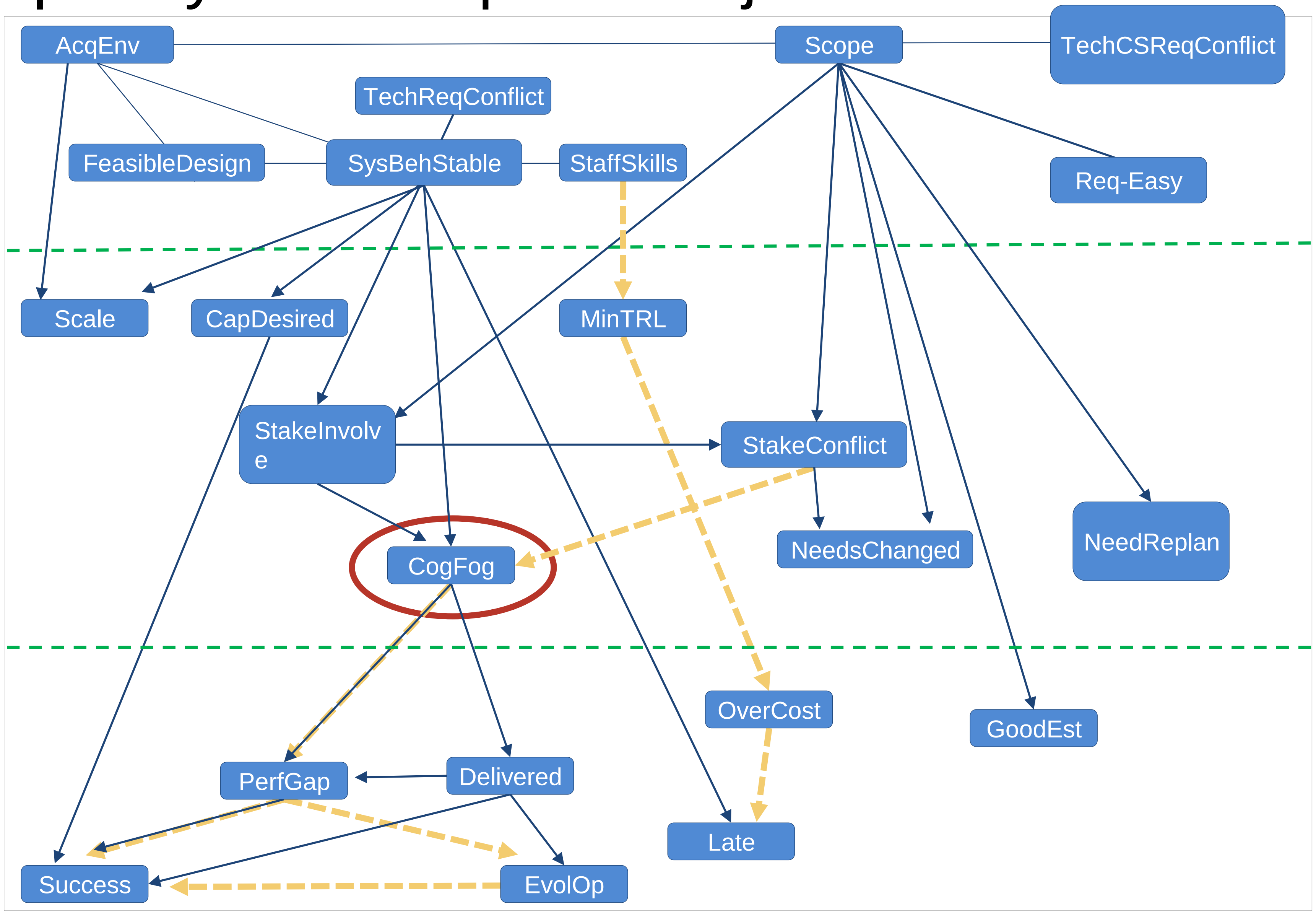
- no evidence for 1)
- consistent evidence of 2) but only mediated through 3)
- consistent evidence for 3)
- weak evidence for other paths to success

Legend

A → B   A **directly causes** B (all other factors held constant; a change in A results in a change to B)

A—B   Either A → B or A ← B, but which one was not determined

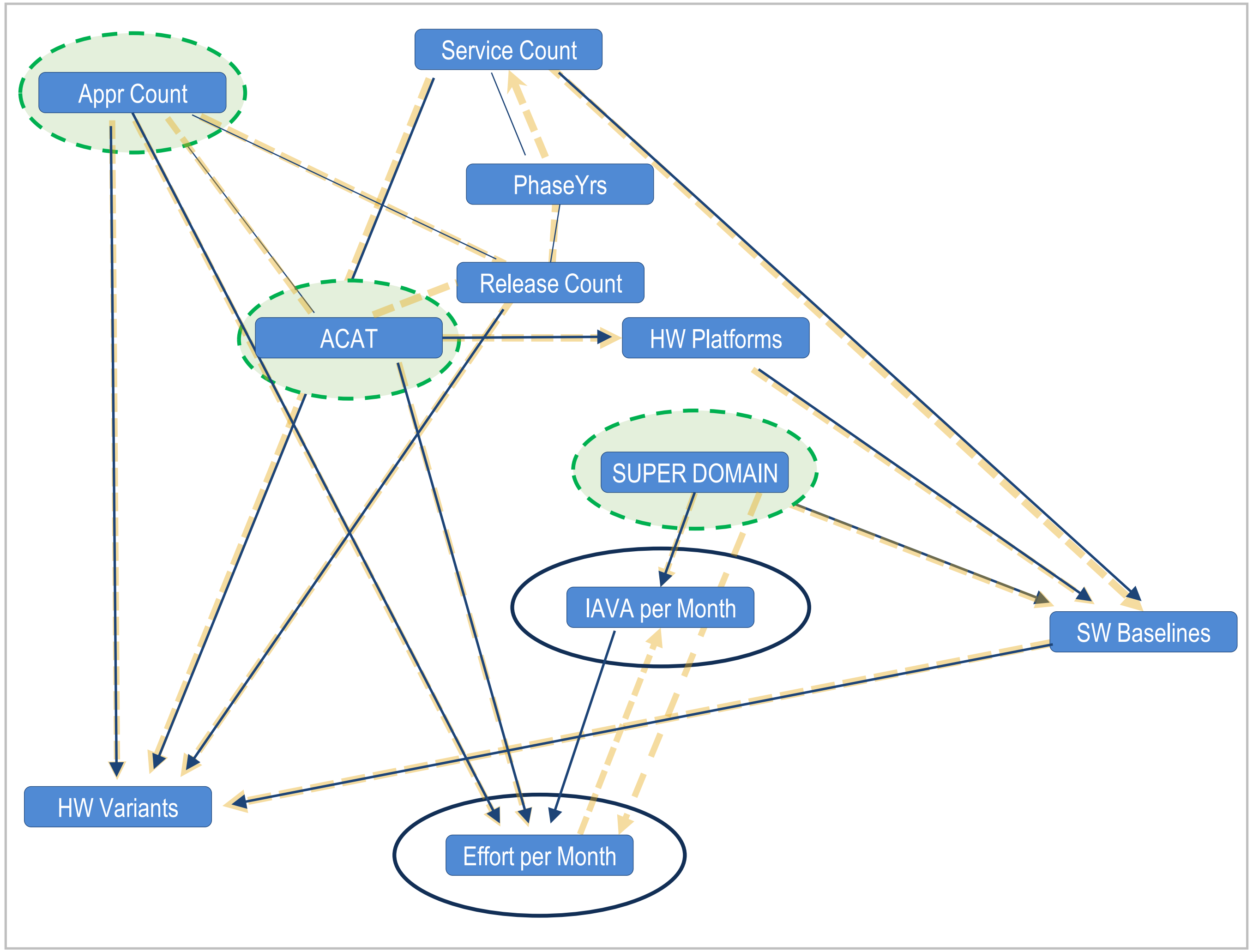
Gold edges represent results from a second search algorithm.



© 2019 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

Which Factors Drive the Number of IAVAs and Effort per Month?



Causal learning found that

1. Super Domain (SD) drives IAVAs and effort (per month).
   - IAVAs and effort are causally related, but their relationship vanishes if data is segmented by SD.
2. The number of appropriations and ACAT also drives effort.
   - This could be interpreted to mean we're missing some controllable measures.
3. Could other measures provide insight?
   - accounting type
   - number of IAVAs opened and closed
   - technical stack

© 2019 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

Do Architecture Pattern Violations Cause Vulnerabilities?

Direction of Causality

Outcome: File Affiliation with Total Security Issues

		Entirety of Chromium	Extensions Partition	UI Partition	Other Partition	Chromeos Partition	Resources Partition
Layer 1 **Exogenous**	Architecture Partition	Green	Grey	Grey	Grey	Grey	Grey
	File Age	Orange	Orange	Orange	Green	Red	Red
	Latest LoC	Orange	Orange	Orange	Orange	Red	Red
Layer 2 **Architecture Pattern Violations**	Clique	Green	Green	Green	Green	Red	Red
	Crossing	Orange	Green	Orange	Orange	Red	Red
	ModularityViolation	Green	Red	Green	Green	Red	Red
	PackageCycle	Orange	Green	Orange	Green	Red	Red
	UnhealthyInheritance	Orange	Red	Orange	Grey	Red	Red
	UnstableInterface	Orange	Orange	Green	Green	Red	Red
Layer 3 **Interim Outcomes**	Bug Churn	Green	Orange	Orange	Green	Red	Red
	CoChange	Green	Green	Green	Green	Red	Red
	NonBug Churn	Green	Green	Orange	Orange	Red	Red
	NonBug Commit	Green	Orange	Grey	Green	Red	Red
	Weighted CoChange	Green	Grey	Grey	Grey	Red	Red
Layer 4 **Final Outcome**	*% of files affiliated with Security Issues*	3.3%	87.8%	3.7%	3.8%	0.4%	1.2%

Legend **Green** = Direct Causal Evidence | **Orange** = Indirect Causal Evidence | **Red** = No Causal Evidence | **Grey** = Not Applicable

© 2019 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

Conclusions and Future Work

Progress in **software engineering** can be accelerated by using **causal learning.**

- identifying deliberate courses of action
  - programmatic decisions and policy formulation
- focusing measurement on factors identified as causally related to outcomes of interest
  - **We may be measuring the wrong things and acting on the wrong signals.**

In the coming year, we will

- investigate determinants and dimensions of quality
- quantify the strength of causal relationships
- seek replication with other data sets and continue to refine our methodology
- integrate the results into a unified set of decision-making principles

We want your help! Stop by our poster or find us to learn how you can get involved.

© 2019 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

Contact Information

Presenters/Points of Contact

Mike Konrad (mdk@sei.cmu.edu)
+1 412.268.5813

Bill Nichols (wrn@sei.cmu.edu)

+1 412.268.1727

Bob Stoddard (rws@sei.cmu.edu)

+1 412.268.1121

Dave Zubrow (dz@sei.cmu.edu)

+1 412.268.5243

Other SEI Team Members

Sarah Sheard

Rhonda Brown

Chris Miller

CMU Contributors

David Danks

Madelyn Glymour

Joe Ramsey

Kun Zhang

USC Contributors

Anandi Hira, Jim Alstad, Barry Boehm

© 2019 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.